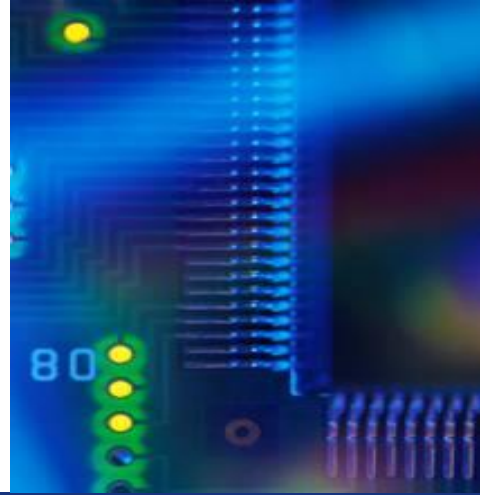
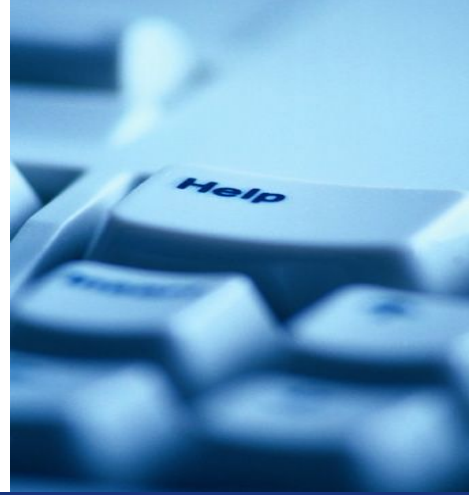




**UNIVERSITAS
BUDI LUHUR**



FAKULTAS TEKNOLOGI INFORMASI

Mobile Programming

[PG119/ 3 SKS]

Riskiana Wulan, MKom

HANDS ON LAB

Building a Basic App with Multiple Screens Using Activity and Intent

Part 1: Setting Up the Project

Create a New Android Project:

- Open **Android Studio**.
- Create a new project:
 - Select **Empty Activity**.
 - Name the project `MultiScreenApp`.
 - Set the language to **Java**.
 - Select **Minimum API Level 21** (Android 5.0 Lollipop).
 - Click **Finish** to set up the project.

Basic App Structure:

- By default, Android Studio creates a `MainActivity.java` file, which will serve as the entry point for your app.

Notes: bisa melanjutkan menggunakan project dari Module Build your first mobile app

Part 2: Adding a Second Activity

Create a New Activity:

- Right-click on the app/src/main/java/your.package.name directory.
- Select **New -> Activity -> Empty Activity**.
- Name the activity SecondActivity.
- This will create SecondActivity.java and its corresponding layout file.

2. Design the Layout for SecondActivity

- Open the XML file for SecondActivity.
- Modify the layout to include a TextView and a Button to navigate back to the main screen.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:app="http://schemas.android.com/apk/res-auto"
        xmlns:tools="http://schemas.android.com/tools"
        android:id="@+id/main"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:context=".SecondActivity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="This is the Second Activity"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.462"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.343" />
```

```
<Button
    android:id="@+id/btnBack"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="back"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.419"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.435" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Part 3: Navigating Between Activities Using Intents

1. Add Button in MainActivity to Open SecondActivity:

- In activity_main.xml, add a button to open the second activity:

```
<Button
    android:id="@+id/btnOpenSecondActivity"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="28dp"
    android:text="Open second Activity"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView2" />
```

2.Handle Button Click to Open SecondActivity:

- In MainActivity.java, write the code to open SecondActivity using an **Intent**:

```
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);

        Button btnOpenSecondActivity= findViewById(R.id.btnOpenSecondActivity);
        btnOpenSecondActivity.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(MainActivity.this, SecondActivity.class);
                intent.putExtra("message", "Hello from MainActivity!");
                startActivity(intent);
            }
        });
    }
}
```

3. Handle "Go Back" Button in SecondActivity:

In `SecondActivity.java`, add a listener for the "Go Back" button to finish the activity and return to the previous screen.

```
public class SecondActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_second);
        Intent intent = getIntent();
        String message = intent.getStringExtra("message");

        TextView textView = findViewById(R.id.textview);
        textView.setText(message);

        Button btnGoBack = findViewById(R.id.btnBack);
        btnGoBack.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                finish(); // Ends the current activity and returns to the previous one
            }
        });
    }
}
```

Passing Data Between activities

1. Pass Data from MainActivity to SecondActivity:

- Modify the **Intent** in MainActivity to send a string message to SecondActivity using **Intent Extras**.

```
btnOpenSecondActivity.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(MainActivity.this, SecondActivity.class);
        intent.putExtra("message", "Hello from MainActivity!");
        startActivity(intent);
    }
});
```

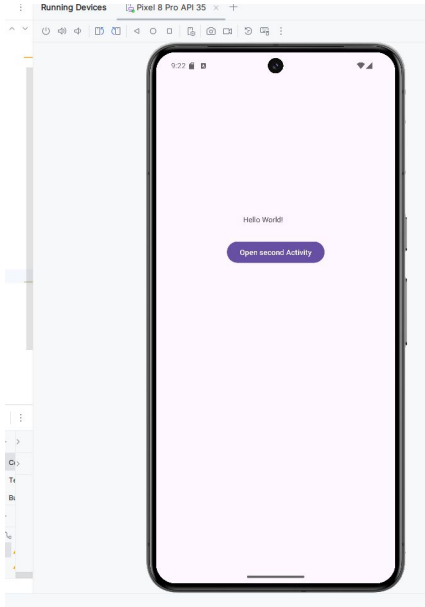
2. Retrieve Data in SecondActivity:

- In `SecondActivity.java`, retrieve the string passed from `MainActivity` using `getIntent().getStringExtra()`.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    EdgeToEdge.enable(this);
    setContentView(R.layout.activity_second);
    Intent intent = getIntent();
    String message = intent.getStringExtra("message");

    TextView textView = findViewById(R.id.textView);
    textView.setText(message);
}
```

Run the app



1. Click open
second Activity

2. Click back
button

